

# 一个 TaskTracker 执行多种任务类型

有些人，使用 LTS 的时候，当有很多类型任务的时候，是不是需要很多个 TaskTracker 去执行这些不同的任务，一种任务是不是一定需要一个 TaskTracker 去对应。

这里我给出参考性的建议，当你某个任务量非常大，或者非常耗时的时候，可以将这个类型的任务独立出一个或多个 TaskTracker 来执行这个。当你任务量不大的时候，可以用一个 TaskTracker 来执行多种类型的任务。那么怎么去做呢。

主要是通过 Job 中自定义参数去实现，可以在 Job 中指定参数 譬如 type, Job.setParam("type", "A"), 那么在执行的时候，可以根据 type 来决定到底是走哪个业务逻辑。下面我给一个参考例子。

```
/**
 * 总入口，在 taskTracker.setJobRunnerClass(JobRunnerDispatcher.class)
 * JobClient 提交 任务时指定 Job 类型 job.setParam("type", "aType")
 * @author Robert HG (254963746@qq.com) on 8/19/14.
 */
public class JobRunnerDispatcher implements JobRunner {

    private static final ConcurrentHashMap<String/*type*/, JobRunner>
        JOB_RUNNER_MAP = new ConcurrentHashMap<String, JobRunner>();

    static {
        JOB_RUNNER_MAP.put("aType", new JobRunnerA()); // 也可以从Spring中拿
        JOB_RUNNER_MAP.put("bType", new JobRunnerB());
    }

    @Override
    public Result run(Job job) throws Throwable {
        String type = job.getParam("type");
        return JOB_RUNNER_MAP.get(type).run(job);
    }
}

class JobRunnerA implements JobRunner {
    @Override
    public Result run(Job job) throws Throwable {
        // TODO A类型Job的逻辑
        return null;
    }
}

class JobRunnerB implements JobRunner {
    @Override
    public Result run(Job job) throws Throwable {
        // TODO B类型Job的逻辑
        return null;
    }
}
```

